

```
--
-- Copyright $Date: 1997/04/11 04:33:25 $ Jtec Pty Limited
--
-- Jtec reserves the right to make changes in
-- specification and other information contained in this document
-- without prior notice. The reader should consult Jtec
-- to determine whether any such changes have been made.
--
-- In no event shall Jtec be liable for any incidental,
-- indirect, special, or consequential damages whatsoever (including
-- but not limited to lost profits) arising out of or related to this
-- document or the information contained in it, even if Jtec
-- has been advised of, known, or should have known, the
-- possibility of such damages.
--
-- Jtec grants vendors, end-users, and other interested parties
-- a non-exclusive license to use this Specification in connection
-- with the management of Jtec products.
--
-- $Author: matthewg $
-- $Revision: 1.11 $
```

```
-----
-- Object (Module):
--   Module Management
-- Status:
--   Optional
-- Description:
--   All management information for Module Management.
-----
```

JTEC-MODULEMANAGEMENT-MIB DEFINITIONS

```
::= BEGIN
```

IMPORTS

```
OBJECT-TYPE
    FROM RFC-1212
TRAP-TYPE
    FROM RFC-1215
TimeTicks
    FROM RFC1155-SMI
jtec, jservices,
    JmoAdminState, JmoCount, JmoTotal, JmoDescription,
    JmoInteger, JmoName, JmoHistory, JmoIdentifier,
    JmoInstance, JmoRowStorage, JmoFilename, JmoSerialNumber,
    JmoVersion, JmoBoolean, JmoTimeStamp, JmoPhysAddress,
    JmoConfigRowStatus, JmoTimeTicks
    FROM JTEC-CORE-MIB
;
```

```
-----
-- Object (Group):
--   Module Management
-- Status:
--   Mandatory
-- Description:
--   Specifies the group that contains all module management
--   information.
```

```
-----  
jmoModuleManagement OBJECT IDENTIFIER  
 ::= { jservices 1 }
```

```
-----  
-- Object (Textual Conventions):  
--   Textual Conventions  
-- Status:  
--   Mandatory  
-- Description:  
--   These are the textual conventions employed by Module  
--   Management.  
-----
```

```
-----  
-- Object (Textual Convention):  
--   ModuleRestartAction  
--  
-- Status:  
--   Mandatory  
--  
-- Description:  
--   This is used to specify what behaviour should occur when  
--   the module manager is restarted. The following values are  
--   defined:  
--  
--   * clean -- start up without any knowledge of existing  
--     modules.  
--  
--   * reinstate-failed-modules -- start up with knowledge  
--     of modules that were in existence prior to shut-  
--     down, and mark those modules as being failed.  
-----
```

```
JmoModuleRestartAction  
 ::= INTEGER { clean (1),  
              reinstate-failed-modules (2)  
              }
```

```
-----  
-- Object (Textual Convention):  
--   ModuleType  
--  
-- Status:  
--   Mandatory  
--  
-- Description:  
--   This is used to identify a particular type of module. The  
--   following values are defined:  
--  
--   * other -- the type of module is unknown, either because  
--     it has not been specified yet, or because it is of a  
--     type not currently represented in the MIB.  
--  
--   * resource -- the module is a "Resource Module".  
--  
--   * combo -- the module is a "Combo Module".  
--  
--   * isdn_primary_rate_te -- the module is a  
--     "ISDN Primary Rate TE Module".  
-----
```

```
--
-- * isdn_primary_rate_nt -- the module is a
--   "ISDN Primary Rate NT Module".
--
-- * isdn_basic_rate_te -- the module is a
--   "ISDN Basic Rate TE Module".
--
-- * isdn_basic_rate_nt -- the module is a
--   "ISDN Basic Rate NT Module".
--
```

```
-----
JmoModuleType
 ::= INTEGER { other (1),
              resource (2),
              combo (3),
              isdn_primary_rate_te (4),
              isdn_primary_rate_nt (5),
              isdn_basic_rate_te (6),
              isdn_basic_rate_nt (7)
            }

```

```
-----
-- Object (Textual Convention):
--   ModuleUploadFilename
--
-- Status:
--   Mandatory
--
-- Description:
--   This is used to specify the filename that will be used to
--   perform an image upload to a module.
--
```

```
-----
JmoModuleUploadFilename
 ::= JmoFilename

```

```
-----
-- Object (Textual Convention):
--   ModuleUploadType
--
-- Status:
--   Mandatory
--
-- Description:
--   This is used to specify the type of upload that should
--   occur for a module. The following values are defined:
--
--   * never -- never perform an upload for the module.
--
--   * always -- always perform an upload for the module.
--
--   * older-version -- perform an upload if the version
--     of software on the module is older than the image
--     version.
--
--   * newer-version -- perform an upload if the version
--     of software on the module is newer than the image
--     version.
--
--   * different-version -- perform an upload if the version
--     of software on the module is different (either newer
--     or older) to the image version.
--
```

--

```
JmoModuleUploadType
 ::= INTEGER { never (1),
              always (2),
              older-version (3),
              newer-version (4),
              different-version (5)
            }
```

-- Object (Textual Convention):
-- ModuleManagerNotifications

-- Status:
-- Mandatory

-- Description:
-- This is used to indicate the type of notifications that
-- should be generated by the module manager. This is a bit
-- string, constructed by logically or'ing any of the values.
-- The none (0) value indicates that none are set. The following
-- values are defined:

- * none -- no notifications are to be generated.
- * state-change -- notify the manager when the state of the
-- module manager occurs.
- * modules-active-threshold -- notify the manager when the
-- number of active modules has exceeded the threshold
-- limit.
- * modules-failed-threshold -- notify the manager when the
-- number of failed modules has exceeded the threshold
-- limit.

```
JmoModuleManagerNotifications
 ::= INTEGER { none (0),
              state-change (1),
              modules-active-threshold (2),
              modules-failed-threshold (4)
            }
```

-- Object (Textual Convention):
-- GenericModuleAddress

-- Status:
-- Mandatory

-- Description:
-- This is used to specify a physical address for a module.

```
JmoGenericModuleAddress
 ::= JmoPhysAddress
```

-- Object (Textual Convention):

```

--      GenericModuleSubType
--
--      Status:
--          Mandatory
--
--      Description:
--          This is used to specify the "subtype" of a module, which it
--          currently does so only for a generic type of module. The
--          following values are defined:
--
--          * other -- the more specified type is unknown.
--
--          * none -- there is no more specified type.
--
-----

```

```

JmoGenericModuleSubType
 ::= INTEGER { other (1),
              none (2)
            }

```

```

-----
--      Object (Textual Convention):
--          GenericModuleState
--
--      Status:
--          Mandatory
--
--      Description:
--          This is used to specify the state of a module. The module
--          will always exist in one of these states. The following
--          values are defined:
--
--          * other -- the state of the module is unknown.
--
--          * inactive -- the module is inactive, in which case it
--            is sitting in the chassis not doing anything.
--
--          * uploading -- the module is uploading, in which case it
--            is transferring an image to the module itself.
--
--          * active -- the module is active, in which case it is
--            operating and presumably initiating or terminating
--            calls.
--
--          * failed -- the module has failed, this may occur due to
--            a physical isolation (e.g. cbus failure), upload
--            problem, or software logic failure.
--
-----

```

```

JmoGenericModuleState
 ::= INTEGER { other (1),
              inactive (2),
              uploading (3),
              active (4),
              failed (5)
            }

```

```

-----
--      Object (Textual Convention):
--          GenericModuleNotifications
--
--      Status:

```

```

--      Mandatory
--
-- Description:
--      This is used to indicate the type of notifications that
--      should be generated by the generic module. This is a bit
--      string, constructed by logically or'ing any of the values.
--      The none (0) value indicates that none are set. The following
--      values are defined:
--
--      * none -- no notifications are to be generated.
--
--      * state-change -- notify the manager when the state of the
--        module occurs.
--
-----

```

```

JmoGenericModuleNotifications
 ::= INTEGER { none (0),
              state-change (1)
            }

```

```

-----
-- Object (Textual Convention):
--      IPModuleSubType
--
-- Status:
--      Mandatory
--
-- Description:
--      This is used to specify the "subtype" of an IPM module. The
--      following values are defined:
--
--      * other -- the more specified type is unknown.
--
--      * ts14 -- Australian standard.
--
--      * etsi -- European standard.
--
--      * dass -- British standard.
--
--      * dpnss -- Digital Private Network Signalling System.
--
-----

```

```

JmoIPModuleSubType
 ::= INTEGER { other (1),
              ts14 (2),
              etsi (3),
              dass (4),
              dpnss (5)
            }

```

```

-----
-- Object (Textual Convention):
--      ElModuleSubType
--
-- Status:
--      Mandatory
--
-- Description:
--      This is used to specify the "subtype" of an ElM module. The
--      following values are defined:
--
--      * other -- the more specified type is unknown.

```

```

--
-----

JmoElModuleSubType
 ::= INTEGER { other (1)
              }

-----

-- Object (Textual Convention):
--   BRModuleSubType
--
-- Status:
--   Mandatory
--
-- Description:
--   This is used to specify the "subtype" of an BRM module. The
--   following values are defined:
--
--   * other -- the more specified type is unknown.
--
--   * ts13 -- Australian standard.
--
--   * etsi -- European standard.
--
--   * spectrum -- .
-----

JmoBRModuleSubType
 ::= INTEGER { other (1),
              ts13 (2),
              etsi (3),
              spectrum (4)
              }

-----

-- Object (Textual Convention):
--   QBRModuleSubType
--
-- Status:
--   Mandatory
--
-- Description:
--   This is used to specify the "subtype" of an QBRM module. The
--   following values are defined:
--
--   * other -- the more specified type is unknown.
--
--   * multiprotocol -- multiple basic rate protocols.
--
--   * ts13 -- Australian standard.
--
--   * etsi -- European standard.
--
--   * spectrum -- .
-----

JmoQBRModuleSubType
 ::= INTEGER { other (1),
              multiprotocol (2),
              ts13 (3),
              etsi (4),
              spectrum (5)
              }

```

```
}
```

```
-----  
-- Object (Group):  
--     Manager << Module Management  
-- Status:  
--     Mandatory  
-- Description:  
--     This group contains management information for high level  
--     module management. This is responsible for overseeing all  
--     individual modules.  
-----
```

```
jmoMmManager OBJECT IDENTIFIER  
 ::= { jmoModuleManagement 1 }
```

```
-----  
-- Object (Item):  
--     Config << Manager  
-- Status:  
--     Mandatory  
-- Description:  
--     This group contains the configuration that is used by the  
--     module manager to perform its activities.  
-- Applicability:  
--     Release 1  
-----
```

```
jmoMmMgrConfig OBJECT IDENTIFIER  
 ::= { jmoMmManager 1 }
```

```
-- State  
jmoMmMgrCfgState OBJECT-TYPE  
     SYNTAX      JmoAdminState  
     ACCESS      read-write  
     STATUS      mandatory  
     DESCRIPTION "Indicates whether module management should be  
                 enabled or disabled. If disabled, then new  
                 modules plugged in are not seen by the Resource  
                 Module."  
     DEFVAL      { enabled }  
 ::= { jmoMmMgrConfig 1 }
```

```
-- Modules Active Max  
-- Applicability: Future  
jmoMmMgrCfgModsActvMax OBJECT-TYPE  
     SYNTAX      JmoCount  
     ACCESS      read-write  
     STATUS      mandatory  
     DESCRIPTION "The maximum number of modules that can be  
                 present in the system at any given point in  
                 time. If this number of modules are already  
                 active in the system, then no more can be  
                 created. 0 specifies no limit."  
     DEFVAL      { 0 }  
 ::= { jmoMmMgrConfig 2 }
```

```
-- Modules Active Threshold  
-- Applicability: Future  
jmoMmMgrCfgModsActvThr OBJECT-TYPE
```

```

SYNTAX      JmoCount
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "Indicates a threshold at which notifications
            should be generated to indicate that an excessive
            number of modules are present in the system. This
            serves as an informative warning to the NMS about
            over load conditions in the system. Specified as
            an absolute number. 0 specifies no limit."
DEFVAL      { 0 }
::= { jmoMmMgrConfig 3 }

-- Modules Failed Threshold
-- Applicability: Future
jmoMmMgrCfgModsFldThr OBJECT-TYPE
    SYNTAX      JmoCount
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates a threshold at which alarms should
                be generated to indicate that an excessive
                number of modules failures has occurred.
                Specified as an absolute number. 0 specifies
                no limit."
    DEFVAL      { 0 }
    ::= { jmoMmMgrConfig 4 }

-- Restart Action
-- Applicability: Future
jmoMmMgrCfgRestartAction OBJECT-TYPE
    SYNTAX      JmoModuleRestartAction
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates whether modules should be reinstated
                from SRAM upon restart. This means that each
                module will be started in a failed condition,
                and notifications will potentially be sent to
                the NMS."
    DEFVAL      { clean }
    ::= { jmoMmMgrConfig 5 }

-- Ignore Unconfigured Modules
-- Applicability: Future
jmoMmMgrCfgIgnoreUncMods OBJECT-TYPE
    SYNTAX      JmoBoolean
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates whether or not modules that are not
                configured should be accepted for use by the
                system. The purpose for this is to allow for
                some level of constraint to be placed upon what
                can and can not operate in the system."
    DEFVAL      { false }
    ::= { jmoMmMgrConfig 6 }

-- History
-- Applicability: Future
jmoMmMgrCfgHistory OBJECT-TYPE
    SYNTAX      JmoHistory
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Specifies what history is to be used for the
                status information."
    DEFVAL      { enabled }
    ::= { jmoMmMgrConfig 7 }

```

```

-- Notifications
-- Applicability: Future
jmoMmMgrCfgNotifications OBJECT-TYPE
    SYNTAX      JmoModuleManagerNotifications
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION  "Specifies what notifications are to be used
                  for the status information."
    DEFVAL      { none }
 ::= { jmoMmMgrConfig 8 }

```

```

-----
-- Object (Item):
--   Status << Manager
-- Status:
--   Mandatory
-- Description:
--   This group contains the status information about the module
--   manager. This is used to determine workload characteristics
--   relating to the activation and failure of modules.
-- Applicability:
--   Release 1
-----

```

```

jmoMmMgrStatus OBJECT IDENTIFIER
 ::= { jmoMmManager 2 }

```

```

-- State
jmoMmMgrStState OBJECT-TYPE
    SYNTAX      JmoAdminState
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION  "Indicates whether module management is enabled
                  or disabled. This can only be changed in the
                  configuration."
 ::= { jmoMmMgrStatus 1 }

```

```

-- State Change Time
jmoMmMgrStStateChngTime OBJECT-TYPE
    SYNTAX      JmoTimeStamp
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION  "Indicates the last time at which module
                  management changed between being enabled or
                  disabled. This cannot be changed."
 ::= { jmoMmMgrStatus 2 }

```

```

-- State Change Reason
jmoMmMgrStStateChngRsn OBJECT-TYPE
    SYNTAX      JmoDescription
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION  "Indicates the reason for the last state change,
                  which will generally indicate that it was manager
                  initiated. This cannot be changed."
 ::= { jmoMmMgrStatus 3 }

```

```

-- State Previous
jmoMmMgrStStatePrev OBJECT-TYPE
    SYNTAX      JmoAdminState
    ACCESS      read-only

```

```
STATUS      mandatory
DESCRIPTION "Indicates the previous state before the last
            change. This cannot be changed."
::= { jmoMmMgrStatus 4 }
```

```
-- Modules Active
-- Applicability: Future
jmoMmMgrStModulesActive OBJECT-TYPE
    SYNTAX      JmoCount
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Indicates the number of modules that are
                currently active, irrespective of the state
                of the module. This cannot be reset."
::= { jmoMmMgrStatus 5 }
```

```
-- Modules Active Max
-- Applicability: Future
jmoMmMgrStModulesActiveMax OBJECT-TYPE
    SYNTAX      JmoCount
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates the maximum number of modules that
                have been concurrently active at any given
                point in time from when the system was started.
                This can be reset by writing a zero."
::= { jmoMmMgrStatus 6 }
```

```
-- Modules Activated
-- Applicability: Future
jmoMmMgrStModulesActivated OBJECT-TYPE
    SYNTAX      JmoTotal
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates the total number of modules that
                have been active since the system was started.
                This can be reset by writing a zero."
::= { jmoMmMgrStatus 7 }
```

```
-- Modules Failed
-- Applicability: Future
jmoMmMgrStModulesFailed OBJECT-TYPE
    SYNTAX      JmoTotal
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates the total number of modules that have
                failed (for internal reasons) since the system
                has started. This can be reset by writing a zero."
::= { jmoMmMgrStatus 8 }
```

```
-- Modules Failed Max
-- Applicability: Future
jmoMmMgrStModulesFailedMax OBJECT-TYPE
    SYNTAX      JmoCount
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates the maximum number of modules that
                have failed from when the system was started.
                This can be reset by writing a zero."
::= { jmoMmMgrStatus 9 }
```

-- Object (Notify):

```
--      Manager State Change Notification << Manager
-- Status:
--      Mandatory
-- Description:
--      This notification is used to specify a state change that
--      may occur in relation to the module manager.
-- Applicability:
--      Future
```

```
jmoMmMgrNotifyStateChange TRAP-TYPE
ENTERPRISE   jtec
VARIABLES    { jmoMmMgrStState,
                jmoMmMgrStStatePrev,
                jmoMmMgrStStateChngTime,
                jmoMmMgrStStateChngRsn }
DESCRIPTION  "This notification is generated when the
              state of the module manager has changed. It
              is used to report the state change to the
              NMS."

 ::= 1000
```

```
-- Object (Notify):
--      Manager Modules Active Notification << Manager
-- Status:
--      Mandatory
-- JmoDescription:
--      This notification is used to inform the manager about
--      modules that have become active.
-- Applicability:
--      Future
```

```
jmoMmMgrNotifyMdlsActThrshld TRAP-TYPE
ENTERPRISE   jtec
VARIABLES    { jmoMmMgrStModulesActive,
                jmoMmMgrCfgModsActvThr }
DESCRIPTION  "This notifications is generated when the
              number of plugged in modules has exceeded
              the threshold of plugged in modules. This
              is used to inform the NMS about numbers of
              plugged in modules."

 ::= 1001
```

```
-- Object (Notify):
--      Manager Modules Failed Notification << Manager
-- Status:
--      Mandatory
-- Description:
--      This notification is used to inform the manager about
--      the number of modules that have failed.
-- Applicability:
--      Future
```

```
jmoMmMgrNotifyModulesFailedThreshold TRAP-TYPE
ENTERPRISE   jtec
VARIABLES    { jmoMmMgrStModulesFailed,
                jmoMmMgrCfgModsFldThr }
DESCRIPTION  "This notifications is generated when the
              number of failures from any module has
              exceeded a defined threshold. This allows the
```

```
                NMS to be alerted to failure conditions."
 ::= 1002
```

```
-----
-- Object (Group):
--   Generic << Module Management
-- Status:
--   Mandatory
-- Description:
--   This group contains management information for generic module,
--   which acts as a base from which specialised modules derive. At
--   this point in time, there are no specialisations on a per
--   module basis.
-----
```

```
jmoMmModuleGeneric OBJECT IDENTIFIER
 ::= { jmoModuleManagement 2 }
```

```
-----
-- Object (Table):
--   Config << Module Generic
-- Status:
--   Mandatory
-- Description:
--   Contains the configuration that is common to all modules.
--   This is the the profile that is assigned to a module upon
--   sign on, through the mapping table.
-----
```

```
jmoMmModGenConfigTable OBJECT-TYPE
 SYNTAX      SEQUENCE OF JmoMmModGenConfigEntry
 ACCESS      not-accessible
 STATUS      mandatory
 DESCRIPTION "Contains all of the configuration required to
             define the operation of a single module."
 ::= { jmoMmModuleGeneric 1 }
```

```
jmoMmModGenConfigTableLength OBJECT-TYPE
 SYNTAX      JmoInteger
 ACCESS      read-only
 STATUS      mandatory
 DESCRIPTION "Get the number of rows in the Table."
 ::= { jmoMmModuleGeneric 2 }
```

```
jmoMmModGenConfigTableIndex OBJECT-TYPE
 SYNTAX      JmoInstance
 ACCESS      read-only
 STATUS      mandatory
 DESCRIPTION "Get an index for a row in the Table."
 ::= { jmoMmModuleGeneric 3 }
```

```
-- Entry
jmoMmModGenConfigEntry OBJECT-TYPE
 SYNTAX      JmoMmModGenConfigEntry
 ACCESS      not-accessible
 STATUS      mandatory
 DESCRIPTION "Entry for the Table."
 INDEX      { jmoMmModGenCfIndex }
 ::= { jmoMmModGenConfigTable 1 }
```

```
JmoMmModGenConfigEntry
```

```

::= SEQUENCE { -- JmoInstance
    jmoMmModGenCfgIndex JmoInstance, -- index
    jmoMmModGenCfgName JmoName,
    -- Administrative Control
    jmoMmModGenCfgState JmoAdminState,
    -- Version Control
    jmoMmModGenCfgVersionMinimum JmoVersion,
    jmoMmModGenCfgVersion JmoVersion,
    -- Upload Information
    jmoMmModGenCfgUploadFilename JmoModuleUploadFilename,
    jmoMmModGenCfgUploadType JmoModuleUploadType,
    -- Management
    jmoMmModGenCfgHistory JmoHistory,
    jmoMmModGenCfgNotifications JmoGenericModuleNotifications,
    -- Row Control
    jmoMmModGenCfgStatus JmoConfigRowStatus,
    jmoMmModGenCfgStorage JmoRowStorage }

-- JmoInstance
jmoMmModGenCfgIndex OBJECT-TYPE
    SYNTAX      JmoInstance
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "This is the index for the profile."
    ::= { jmoMmModGenConfigEntry 1 }

-- JmoName
jmoMmModGenCfgName OBJECT-TYPE
    SYNTAX      JmoName
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "A textual name used to identify the profile.
                This corresponds to the profile name."
    ::= { jmoMmModGenConfigEntry 2 }

-- State
jmoMmModGenCfgState OBJECT-TYPE
    SYNTAX      JmoAdminState
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates whether the module is enabled or
                not. If disabled, then it is not made visible
                to the Resource Module."
    DEFVAL      { enabled }
    ::= { jmoMmModGenConfigEntry 3 }

-- Version Minimum
jmoMmModGenCfgVersionMinimum OBJECT-TYPE
    SYNTAX      JmoVersion
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The minimum version of this module to accept
                when signing on. If a version lower than this
                signs on, then the module is ignored. The
                purpose of this is to ensure that incompatible
                modules are not used with the software."
    DEFVAL      { 0 }
    ::= { jmoMmModGenConfigEntry 4 }

-- Version
jmoMmModGenCfgVersion OBJECT-TYPE
    SYNTAX      JmoVersion

```

```

ACCESS      read-write
STATUS      mandatory
DESCRIPTION "The version used to as a reference when
            determining if a module requires a software
            image upload. It is used in conjunction with the
            upload type."
DEFVAL      { 0 }
::= { jmoMmModGenConfigEntry 5 }

-- Upload Filename
jmoMmModGenCfgUploadFilename OBJECT-TYPE
SYNTAX      JmoModuleUploadFilename
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "The filename of the image to upload when the
            module signs on. This will only occur if the
            version indicated in the sign on is less than
            that of the upload filename."
DEFVAL      { "None" }
::= { jmoMmModGenConfigEntry 6 }

-- Upload Type
jmoMmModGenCfgUploadType OBJECT-TYPE
SYNTAX      JmoModuleUploadType
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "Indicates whether or not the upload is mandatory,
            version dependant, or not occurring. This means
            that a module upload can be stopped from
            occurring."
DEFVAL      { never }
::= { jmoMmModGenConfigEntry 7 }

-- History
-- Applicability: Future
jmoMmModGenCfgHistory OBJECT-TYPE
SYNTAX      JmoHistory
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "Indicates whether or not historical info is
            going to be retained for this module."
DEFVAL      { disabled }
::= { jmoMmModGenConfigEntry 8 }

-- Notifications
-- Applicability: Future
jmoMmModGenCfgNotifications OBJECT-TYPE
SYNTAX      JmoGenericModuleNotifications
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "Indicates which notifications are enabled
            for the module."
DEFVAL      { none }
::= { jmoMmModGenConfigEntry 9 }

-- Row Status
jmoMmModGenCfgStatus OBJECT-TYPE
SYNTAX      JmoConfigRowStatus
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "The status of the row."
::= { jmoMmModGenConfigEntry 10 }

-- Row Storage

```

```

jmoMmModGenCfgStorage OBJECT-TYPE
    SYNTAX      JmoRowStorage
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The storage of the row."
    DEFVAL      { persistent }
    ::= { jmoMmModGenConfigEntry 11 }

```

```

-----
-- Object (Table):
--     Mapping Config << Module Generic
-- Status:
--     Mandatory
-- Description:
--     This is the configuration used to associate a module with a
--     profile upon sign on. The module is also given a logical name
--     for identification and management purposes.
-----

```

```

jmoMmModGenMappingConfigTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF JmoMmModGenMappingConfigEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Contains all of the configuration used to
                associate a logical configuration profile
                with a physical module."
    ::= { jmoMmModuleGeneric 4 }

```

```

jmoMmModGenMappingConfigTableLength OBJECT-TYPE
    SYNTAX      JmoInteger
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Get the number of rows in the Table."
    ::= { jmoMmModuleGeneric 5 }

```

```

jmoMmModGenMappingConfigTableIndex OBJECT-TYPE
    SYNTAX      JmoInstance
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Get an index for a row in the Table."
    ::= { jmoMmModuleGeneric 6 }

```

```

-- Entry
jmoMmModGenMappingConfigEntry OBJECT-TYPE
    SYNTAX      JmoMmModGenMappingConfigEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Entry for the Table."
    INDEX      { jmoMmModGenMapCfgIndex }
    ::= { jmoMmModGenMappingConfigTable 1 }

```

```

JmoMmModGenMappingConfigEntry
    ::= SEQUENCE {
        -- JmoInstance
        jmoMmModGenMapCfgIndex JmoInstance, -- index
        -- Identification
        jmoMmModGenMapCfgAddress JmoGenericModuleAddress,
        jmoMmModGenMapCfgType JmoModuleType,
        jmoMmModGenMapCfgSubType JmoGenericModuleSubType,
        -- Name and Profile
        jmoMmModGenMapCfgName JmoName,
        jmoMmModGenMapCfgProfile JmoName,

```

```

-- Row Control
jmoMmModGenMapCfgStatus JmoConfigRowStatus,
jmoMmModGenMapCfgStorage JmoRowStorage }

-- JmoInstance
jmoMmModGenMapCfgIndex OBJECT-TYPE
    SYNTAX      JmoInstance
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "This is the index for the profile."
    ::= { jmoMmModGenMappingConfigEntry 1 }

-- Address
jmoMmModGenMapCfgAddress OBJECT-TYPE
    SYNTAX      JmoGenericModuleAddress
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates the physical card slot or board
                 that this mapping profile is for."
    ::= { jmoMmModGenMappingConfigEntry 2 }

-- Type
jmoMmModGenMapCfgType OBJECT-TYPE
    SYNTAX      JmoModuleType
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates the type of module that this
                 profile is for."
    ::= { jmoMmModGenMappingConfigEntry 3 }

-- Sub Type
jmoMmModGenMapCfgSubType OBJECT-TYPE
    SYNTAX      JmoGenericModuleSubType
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates the sub type of module that this
                 profile is for."
    ::= { jmoMmModGenMappingConfigEntry 4 }

-- Name
jmoMmModGenMapCfgName OBJECT-TYPE
    SYNTAX      JmoName
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates the name assigned to the module."
    ::= { jmoMmModGenMappingConfigEntry 5 }

-- Profile
jmoMmModGenMapCfgProfile OBJECT-TYPE
    SYNTAX      JmoName
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "Indicates the profile assigned to the
                 module."
    DEFVAL      { "Default" }
    ::= { jmoMmModGenMappingConfigEntry 6 }

-- Status
jmoMmModGenMapCfgStatus OBJECT-TYPE
    SYNTAX      JmoConfigRowStatus
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The status of the row."
    ::= { jmoMmModGenMappingConfigEntry 7 }

```

```

-- Storage
jmoMmModGenMapCfgStorage OBJECT-TYPE
    SYNTAX      JmoRowStorage
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The storage of the row."
    DEFVAL      { persistent }
    ::= { jmoMmModGenMappingConfigEntry 8 }

```

```

-----
-- Object (Table):
--     Status << Module Generic
-- Status:
--     Mandatory
-- Description:
--     This is the status information about an individual module.
-----

```

```

jmoMmModGenStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF JmoMmModGenStatusEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "This contains all of the status information
                about a particular module. It is feasible that
                additional status information will be provided
                at a later date on a per module basis."
    ::= { jmoMmModuleGeneric 7 }

```

```

jmoMmModGenStatusTableLength OBJECT-TYPE
    SYNTAX      JmoInteger
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Get the number of rows in the Table."
    ::= { jmoMmModuleGeneric 8 }

```

```

-- Entry
jmoMmModGenStatusEntry OBJECT-TYPE
    SYNTAX      JmoMmModGenStatusEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Entry for the Table."
    INDEX      { jmoMmModGenStIndex }
    ::= { jmoMmModGenStatusTable 1 }

```

```

JmoMmModGenStatusEntry
    ::= SEQUENCE {
        -- JmoInstance
        jmoMmModGenStIndex JmoInstance, -- index
        jmoMmModGenStAddress JmoGenericModuleAddress,
        -- Identification
        jmoMmModGenStName JmoName,
        jmoMmModGenStType JmoModuleType,
        jmoMmModGenStSubType JmoGenericModuleSubType,
        jmoMmModGenStVersion JmoVersion,
        -- Configuration
        jmoMmModGenStProfile JmoName,
        -- State Information
        jmoMmModGenStState JmoGenericModuleState,
        jmoMmModGenStStateChangeTime JmoTimeStamp,
        jmoMmModGenStStateChangeRsn JmoDescription,
        jmoMmModGenStStatePrevious JmoGenericModuleState,
        jmoMmModGenStEventDesc JmoDescription,
    }

```

```

        jmoMmModGenStIdentifier JmoIdentifier,
        -- Version Information
        jmoMmModGenStHwSerNo JmoSerialNumber,
        jmoMmModGenStHwVersion JmoVersion,
        jmoMmModGenStSwVersion JmoVersion,
        jmoMmModGenStDesc JmoDescription,
        -- State Change Information
        jmoMmModGenStSignOnTime JmoTimeStamp,
        jmoMmModGenStSignedOn JmoTotal,
        jmoMmModGenStSignedOff JmoTotal,
        jmoMmModGenStFailed JmoTotal,
        -- Management Information
        jmoMmModGenStHistory JmoHistory,
        jmoMmModGenStNotifications JmoGenericModuleNotifications }

-- Index
jmoMmModGenStIndex OBJECT-TYPE
    SYNTAX      JmoInstance
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the index for the profile."
    ::= { jmoMmModGenStatusEntry 1 }

-- Address
jmoMmModGenStAddress OBJECT-TYPE
    SYNTAX      JmoGenericModuleAddress
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the physical address of the
                module as currently plugged in, it
                corresponds to a card slot, and possibly
                some form of other board identifier."
    ::= { jmoMmModGenStatusEntry 2 }

-- Name
jmoMmModGenStName OBJECT-TYPE
    SYNTAX      JmoName
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the name assigned to the module
                from the mapping profile. It identifies
                the module for management purposes and
                informational purposes."
    ::= { jmoMmModGenStatusEntry 3 }

-- Type
jmoMmModGenStType OBJECT-TYPE
    SYNTAX      JmoModuleType
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the type of the module, as learned
                through the module's sign on procedures."
    ::= { jmoMmModGenStatusEntry 4 }

-- Sub Type
jmoMmModGenStSubType OBJECT-TYPE
    SYNTAX      JmoGenericModuleSubType
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the subtype of the module, as learned
                through the module's sign on procedures."
    ::= { jmoMmModGenStatusEntry 5 }

```

```

-- Version
jmoMmModGenStVersion OBJECT-TYPE
    SYNTAX      JmoVersion
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the version of the module, as learned
                through the module's sign on procedures."
 ::= { jmoMmModGenStatusEntry 6 }

-- Profile
jmoMmModGenStProfile OBJECT-TYPE
    SYNTAX      JmoName
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the name of the configuration profile
                that has been assigned to the module. It is
                determined through the mapping profile."
 ::= { jmoMmModGenStatusEntry 7 }

-- State
jmoMmModGenStState OBJECT-TYPE
    SYNTAX      JmoGenericModuleState
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "This is the current state of the module. This
                can be changed to explicitly request that the
                module does something (e.g. upload)."
 ::= { jmoMmModGenStatusEntry 8 }

-- State Change Time
jmoMmModGenStStateChangeTime OBJECT-TYPE
    SYNTAX      JmoTimeStamp
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the time at which the state changed
                to its current value. This is used to
                indicate when failures or other situations
                have occurred."
 ::= { jmoMmModGenStatusEntry 9 }

-- State Change Reason
jmoMmModGenStStateChangeRsn OBJECT-TYPE
    SYNTAX      JmoDescription
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the reason for the last state
                change, which may indicate a more precise
                failure or event description."
 ::= { jmoMmModGenStatusEntry 10 }

-- State Previous
jmoMmModGenStStatePrevious OBJECT-TYPE
    SYNTAX      JmoGenericModuleState
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the previous state of the module,
                until the last state change time."
 ::= { jmoMmModGenStatusEntry 11 }

-- Event Description
jmoMmModGenStEventDesc OBJECT-TYPE
    SYNTAX      JmoDescription
    ACCESS      read-only
    STATUS      mandatory

```

```

        DESCRIPTION "This is a description of the last event
                    that may have occurred in association with
                    the module."
 ::= { jmoMmModGenStatusEntry 12 }

-- Identifier
jmoMmModGenStIdentifier OBJECT-TYPE
    SYNTAX      JmoIdentifier
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the identifier that has been assigned
                to the module. It uniquely identifies this
                status when placed into historical storage."
 ::= { jmoMmModGenStatusEntry 13 }

-- Hardware Serial Number
jmoMmModGenStHwSerNo OBJECT-TYPE
    SYNTAX      JmoSerialNumber
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the serial number of the hardware on
                the module, as determined through sign on
                procedures."
 ::= { jmoMmModGenStatusEntry 14 }

-- Hardware Version
jmoMmModGenStHwVersion OBJECT-TYPE
    SYNTAX      JmoVersion
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the revision, or version, of the
                hardware on the module. This is determined
                through the sign on, and may be used to
                discriminate software images."
 ::= { jmoMmModGenStatusEntry 15 }

-- Software Version
jmoMmModGenStSwVersion OBJECT-TYPE
    SYNTAX      JmoVersion
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the build, or version, of the
                software on the module. This is used to
                assist in the upload of new software
                images."
 ::= { jmoMmModGenStatusEntry 16 }

-- Description
jmoMmModGenStDesc OBJECT-TYPE
    SYNTAX      JmoDescription
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is a free form text field that may
                convey additional information about the
                module, such as build and feature
                options."
 ::= { jmoMmModGenStatusEntry 17 }

-- Sign On Time
jmoMmModGenStSignOnTime OBJECT-TYPE
    SYNTAX      JmoTimeStamp
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the time at which the module

```

```

        provided a sign on to the system."
 ::= { jmoMmModGenStatusEntry 18 }

-- Signed On
-- Applicability: Future
jmoMmModGenStSignedOn OBJECT-TYPE
    SYNTAX      JmoTotal
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "This is the total number of sign ons
                that have been provided by the module to
                the system."
 ::= { jmoMmModGenStatusEntry 19 }

-- Signed Off
-- Applicability: Future
jmoMmModGenStSignedOff OBJECT-TYPE
    SYNTAX      JmoTotal
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "This is the total number of sign offs
                that have been provided by the module to
                the system. This can be reset by writing
                a zero."
 ::= { jmoMmModGenStatusEntry 20 }

-- Failed
-- Applicability: Future
jmoMmModGenStFailed OBJECT-TYPE
    SYNTAX      JmoTotal
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "This is the total number of failures that
                have been seen by the system from this
                module. This can be reset by writing a
                zero."
 ::= { jmoMmModGenStatusEntry 21 }

-- History
-- Applicability: Future
jmoMmModGenStHistory OBJECT-TYPE
    SYNTAX      JmoHistory
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "This is the history of the row."
 ::= { jmoMmModGenStatusEntry 22 }

-- Notifications
-- Applicability: Future
jmoMmModGenStNotifications OBJECT-TYPE
    SYNTAX      JmoGenericModuleNotifications
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "These are notifications enabled for this
                module."
 ::= { jmoMmModGenStatusEntry 23 }

```

```

-----
-- Object (Notify):
--   Module State Change << Module Generic
-- Status:
--   Mandatory
-- Description:

```

```
--      These are the notifications that can be generated about a
--      condition associated with a module.
-- Applicability:
--      Future
```

```
-- State Change
```

```
jmoMmModGenNotifyStateChange TRAP-TYPE
```

```
    ENTERPRISE jtec
```

```
    VARIABLES { jmoMmModGenStState,
                jmoMmModGenStStatePrevious,
                jmoMmModGenStStateChangeTime,
                jmoMmModGenStStateChangeRsn,
                jmoMmModGenStType,
                jmoMmModGenStSubType,
                jmoMmModGenStAddress }
```

```
    DESCRIPTION "Indicates that a state change has occurred
                 for the particular module. It provides state
                 and identification information."
```

```
::= 1003
```

```
END -- JTEC-MODULEMANAGER-CORE-MIB
```